AMENDMENTS TO THE CLAIMS

1. (Currently Amended) An apparatus comprising:

a predictor having a collision history table (CHT), said predictor for predicting silent store instructions;

a processing section coupled to the predictor, the processing section including an extended load buffer coupled to the predictor;

a marking processing section;

a comparing processing section; and

a recovery processing section, wherein unexecuted load instructions are advanced over silent store instructions,

wherein the predictor compares an unexecuted load instruction value with an issued and unexecuted store instruction value, and the unexecuted load instruction bypasses the issued store instruction for execution if the unexecuted load instruction value and the issued and unexecuted store value are the same.

2. (Original) The apparatus of claim 1, wherein the predictor is a silent store predictor.

3. (Original) The apparatus of claim 2, wherein the silent store predictor uses path based indexing and the path is based on branches.

4. (Original) The apparatus of claim 3, wherein the silent store predictor is coupled with a state machine.

5. (Original) The apparatus of claim 4, wherein the state machine is one of a 1-bit, a 2-bit and a sticky bit.

6. (Original) The apparatus of claim 1, wherein the predictor is memory dependent.

7. (Original) The apparatus of claim 1, wherein the extended load buffer comprises bit fields to mark load address match, load data match, load predict, and load flush, and bit fields for load address, load attribute and load data.

8. (Original) The apparatus of claim 1, wherein the CHT is one of indexed by a tag and tagless.

9. (Original) The apparatus of claim 1, wherein the CHT includes distance bits.

10. (Currently Amended) A system comprising:

a processor having internal memory,

a bus coupled to the processor;

a memory coupled to a memory controller and the processor;

wherein the processor includes:

a predictor having a collision history table (CHT), said predictor for predicting silent store instructions;

an extended load buffer coupled to the predictor;

a marking process;

a comparing process; and

a recovery process,

wherein unexecuted load instructions are advanced over store instructions, and the predictor compares an unexecuted load instruction value with an issued and unexecuted store instruction value, and the unexecuted load instruction bypasses the issued store instruction for execution if the unexecuted load instruction value and the issued and unexecuted store value are the same.

11. (Original) The system of claim 10, wherein the predictor is a silent store predictor.

12. (Original) The system of claim 11, wherein the silent store predictor uses path based indexing and the path is based on branches.

13. (Original) The system of claim 12, wherein the silent store predictor is coupled with a state machine.

14. (Original) The system of claim 13, wherein the state machine is one of a 1-bit, a 2-bit and a sticky bit.

15. (Original) The system of claim 10, wherein the predictor is memory dependent.

16. (Original) The system of claim 10, wherein the extended load buffer comprises

bit fields to mark load address match, load data match, load predict, and load flush, and bit fields for load address, load attribute and load data.

17. (Original) The system of claim 10, wherein the CHT is one of indexed by a tag and tagless.

18. (Original) The system of claim 10, wherein the CHT includes distance bits.

19. (Currently Amended) A method comprising:

fetching an instruction and determining if an instruction is one of a store and a load;

performing a silent store prediction if the instruction is a store;

~~executing~~ issuing the store instruction;

comparing an address and data of the store instruction with load instructions in an extended load buffer;

setting marking bits in the extended load buffer if a match is found in the comparing;

updating a memory with the store instruction if the store instruction can be retired; and

bypassing a predicted silent store instruction if an unexecuted load instruction value matches the issued and unexecuted store instruction value and executing ~~a~~ the load instruction ~~if the instruction is a load~~ ahead of the predicted silent store instruction.

20. (Currently Amended) The method of claim 19, further comprising preparing ~~a~~ the executed load instruction for retirement if the load instruction is complete, and determining if the load instruction is marked flush in the extended load buffer.

21. (Original) The method of claim 19, wherein the predicting includes marking bits in a collision history table (CHT).

22. (Original) The method of claim 19, wherein the memory is a cache.

23. (Currently Amended) A program storage device readable by a machine comprising instructions that cause the machine to:

fetch an operation and determining if the operation is one of a store instruction and a load instruction;

perform a silent store prediction if the operation is a store instruction;

execute the store operation;

compare an address and data of the store operation with load operations in an extended load buffer;

set marking bits in the extended load buffer if a match is found in the compare instruction;

update a memory with a store operation if the store operation can be retired; and

bypass a predicted silent store operation and execute a load operation ahead of the silent store operation if the operation is a load and the load operation includes a value that matches a value included in the store operation.

24. (Currently Amended) The program storage device of claim 23, wherein the instructions further cause the machine to prepare the load operation for

retirement if ~~a~~ the load ~~instruction~~ operation is complete, and determining if the load operation is marked flush in the extended load buffer.

25. (Original) The program storage device of claim 23, wherein the instruction that causes the machine to predict silent stores includes an instruction that causes the machine to mark bits in a collision history table (CHT).

26. (Original) The program storage device of claim 23, wherein the memory is a cache.